

FEATURE CENTRIC RELEASE MANAGER METHOD AND SYSTEM

FIELD OF THE INVENTION

The invention relates to a method of and system for managing a business process, that is, the product release cycle of an enterprise, including monitoring the operation of the project or product team, evaluating the allocation of resources, scheduling tasks, and assigning tasks to individuals and groups to accomplish the project.

BACKGROUND

In order to accomplish a project, such as a new product introduction, it is necessary to effectively plan, schedule, and monitor progress on many sub-elements. This is the Release Management process. As shown in FIGURE 1, Release Management typically consists of four main stages:

Stage 1: Product marketing defines product features and develops requirements for the product features. The requirements may take the form of requirements documents.

Stage 2: Engineering enumerates tasks to implement the features, sets milestones for the tasks, and completes the tasks to implement the features.

Stage 3: Quality assurance (QA) verifies the engineering designs and the engineering implementations, and tests the features in the product as it is developed.

Stage 4: Other ancillary steps must be completed, for example, the technical publications group documents the features for the end user, the training group may have to develop training programs, there may be business clearances, such as business practices, training, legal, insurance, environmental, marketing, and executive approvals.

Understandably, the product release process involves a lot of information flowing within and between organizations to ensure a quality product is built on time to the right requirements and with proper documentation. Prior project management methods and paradigms having typically emphasized the linearity and linear dependence aspects of Release Management. FIGURE 1 illustrates the apparent linearity of Release Management. Without an integrated system providing overall visibility to help track and communicate task level and sub-task level information, organizations must get by with disconnected systems such as makeshift spreadsheets, white boards, memos, and slips of paper. Often, information is lost or miscommunicated, resulting in a misallocation of resources; missing, undocumented, or inadequately tested features; late product shipments; and, ultimately, dissatisfied customers. And the problem only becomes worse as companies grow and product releases become more frequent and more feature-rich.

Organizations sometimes hire a Release Manager to coordinate the information flow, track release status, and guide important release-related decision making.

SUMMARY

The invention is a method and system for monitoring the development and release of a product, where the product is characterized by having a plurality of features. Typically, each of the features requires a plurality of tasks, as engineering tasks, quality assurance tasks, and technical documentation tasks for approval and completion. The method steps, which the system is configured to carry out, include enumerating features to be included in the product, enumerating tasks, task milestones, task milestone completions, task approvals and feature approvals, and linking them to the features. The tasks are typically grouped by owning function, as engineering, quality assurance, and technical documentation, among others. The enumeration preferably includes information to show linkages, associations, priorities, milestones, and missed milestones. "Link", "linkage", and "linking" are used herein in the sense of a directed logical or

virtual connection, exemplified by links in a linked list database, by integrity constraints and keys in a relational database, and by references in a spreadsheet.

The enumeration of the features, tasks, task milestones, task milestone completions, approvals, and completed approvals, optionally with associated activities, and completed associated activities, is stored in a repository, as a database, for example, a relational database with keys and integrity constraints as the links, or a linked list database, or a spreadsheet. In this way, a first set of entries may correspond to all tasks, task milestones, approvals, and associated activities, and a second set of entries may correspond to task milestone completions, completed approvals, and completed associated activities. When the database is a relational database, each of the features has a primary key, and every task, feature, feature milestone, and approval corresponds to a primary key of a feature and has a unique key of its own.

THE FIGURES

The Figures illustrate aspects of the method and system of the invention.

FIGURE 1 is a time chart, illustrative of both the prior art and the present invention, generally, and showing the direction of order of the feature definition, feature implementation, feature test, and feature documentation steps in the software product release cycle.

FIGURE 2 is a hub-and-spoke illustration of the feature centric paradigm supported by the method and system of the invention.

FIGURE 3 is an expanded hub and spoke illustration of the feature centric paradigm supported by the method and system of the invention

FIGURE 4 illustrates the flow of information between product marketing at the hub, engineering, quality assurance, and technical publications, in the software product release cycle.

[illegible]

FIGURE 12 illustrates the relational database schema of FIGURE 11 interconnected as shown by the relationships of FIGURES 1 and 2, and linked through the PROJECT ID, and the concatenation of the PROJECT ID and the FEATURE ID, with a unique identifier of the concatenation of the PROJECT ID, the FEATURE ID, and the task specific RELEASE ID.

circumference, and joined to the feature, 11, by virtual spokes, 27, (representation information flows) and to each other by another set of virtual links.

no A1
~~FIGURE 3 illustrates as a further application an expanded hub and spoke, feature centric model, deriving from a project, 1. Two features, 11a, and 11b, are illustrated, with engineering tasks, 21a, and 21b, quality assurance tasks, 23a, and 23b, and technical publications tasks, 25a, and 25b. To be noted are the engineering sub tasks, 21ab, 21aa, 21ba, and 21bb.~~ *A1*

The general timeline and sequence of the product release process generally is shown in FIGURE 1, where product marketing begins the release cycle by entering features into the system, determining which are to be implemented, and developing requirements documents. Engineering then defines the tasks required to implement the features and uses the system to track task completion. QA defines test plans to be run against the new features and uses the Release Manager method and system described below to track the test plan development. QA then executes the test plan in various operating environments and records in a database associated with the Release Manager the details of each test iteration. Finally, technical publications documents each feature and uses the Release Manager and its associated database or databases to track document development status.

no A2
~~As shown in FIGURE 3, all engineering tasks, 43, QA tests, 45, and technical publications, 47, are linked back to their associated features. With this feature-centric system, critical decisions such as feature prioritization, resource allocation, and test planning are easily justifiable. Moreover, any Release Manager user can easily determine the development, testing, and documentation status for a given feature. When each of these areas is 100% complete for each feature, the product is ready to ship.~~ *A2*

The feature centric, "hub and spoke" paradigm provides an overall visibility mechanism that empowers a user to communicate detailed feature descriptions throughout the organization, to prioritize features intelligently so that the right features get implemented, to develop marketing requirements documents for engineers to use as product specifications, to allocate engineering

resources and track engineering tasks, to track QA test plan development and test execution, to track documentation development, to ensure release integrity, and to print status reports.

While the invention is described and illustrated with respect to the software product release process, it is, of course, to be understood that the method and system may be readily applied to the product release of any product.

DETAILED DESCRIPTION

The invention provides a method and system for monitoring and managing the product release process using an overall visibility mechanism that positively links features and tasks using, for example, the integrity constraints and keys of the relational database paradigm or, equivalently, the entity-relationship model of other database paradigms. Under the relational database paradigm, used solely for illustration and not limitation, a feature has a unique integrity constraint or master key. Every task or activity necessary to release the feature is positively linked to the feature through the feature's integrity constraint or key.

The invention may be understood by considering the user interfaces. In one embodiment the Release Manager has a set of views that users throughout the organization can use to track and manage the release of products, for example, software products. Although certain organizations will primarily be interested in those views relating directly to them (e.g., engineering will be most interested in the Engineer Task views), all views are available to all organizations to promote information sharing and teamwork. Much of the benefit of Release Manager lies in the linking of data managed by different organizations through a database structure that provides links. For example, product managers will be interested in the engineering tasks underway to implement their features, and QA managers will be interested in the new features that need to be incorporated in test plans. In the Release Manager, features are linked to QA test plans, engineering tasks, and tech pubs documents.

The first step in using the Release Manager is to enter features into the Feature List view of FIGURE 5. This is done by filling in the following fields:

1. Release field 51: This field identifies the product.
2. Title field 53: This field displays a short title for the feature.
3. Description field 55: This field displays a description of the feature.
4. Assoc. Parties field 57: This field associates one or more members of the project team with this feature.
5. Parent field 59: If so desired, features may be organized into a hierarchy of parent features and subfeatures. To designate the current feature as a subfeature of a parent feature, select the parent feature from the picklist.
6. PM Group field 61: This feature associates one or more product marketing groups with this feature.
7. Area field 63: This feature associates a product or feature with a product area or a sub-area or both.
8. Source field 65: This field associates features with sources of the features. Possible sources of features include contractual requirements and market trends.
9. Account field 67: A feature may be the result of a request from or a contract with a specific customer. This field associates the feature with the customer.
10. Revenue field 69: If revenue is tied to delivering this feature, a revenue range may be selected from the list of values.
11. Priority field 71: The priority of the feature and/or the task is displayed in this field.

12. Status field 73: This field displays the overall status of this feature.

13. Related Release Items field 75: If desired, a user may link Marketing Requirements Documents, engineering tasks, QA test plans, and tech docs to this feature through these controls. It is assumed, for purposes of these instructions, that such linking will occur from other views.

In addition, comments may be added and files may be attached to this feature

Marketing Requirements Documents

Product marketing managers enter and track information relating to Marketing Requirements Documents in the Marketing Requirements Document view of FIGURE 6.

The Marketing Requirements Document of FIGURE 6 contains the following fields.

1. Release field 51: This field identifies the product release.
2. Title field 53: This field contains a short title for the Marketing Requirements Document.
3. Description field 55: This field contains a description of the Marketing Requirements Document.
4. Assoc. Parties field 57: This field is used to associate one or more members of the project team with this Marketing Requirements Document.
5. PM Group field 61: This field associates one or more product marketing groups with this Marketing Requirements Document.
6. Status field XX: This field displays the overall status of this Marketing Requirements Document.

7. Rel. Feature field 77: This field is used to link the features documented in this Marketing Requirements Document to a particular record.

Engineering

Engineering managers can use the Engineer Task List view shown in FIGURE 7 to enter and track information relating to tasks being completed to implement features. Engineering tasks may include concept design, detailed design, design verification, prototyping, debugging, limited testing, and, in the case of hardware, prototyping, and trial or pilot production. Information about a task is entered into the "Task Description" area, 79, and information about the progress of the engineering effort is updated in the "Task Status" area, 81. Engineering managers are enabled to work with product marketing to ensure their engineering tasks are properly linked to features they are implementing. Engineering managers enter engineering tasks into the Engineering Task List view by filling in the following fields:

1. Release field 51: Identifies the engineering task to a product.
2. Title field 53: This field displays a short title for the engineering task.
3. Description field 79: This field displays a description of the engineering task.
4. Assoc. Parties field 57: This field associates one or more members of the project team with this task.
5. Eng Group field 83: This field associates one or more engineering groups with this task.
6. Parent field 85: If so desired, engineering tasks may be organized into a hierarchy of parent tasks and subtasks. This field is used to designate the current task as a subtask of a parent task.

7. Priority field 87: In this read-only field, the priority of the engineering task is calculated automatically as the highest priority of all linked features.

8. Effort field 89: This field displays the level of effort required to complete the task from the list of values.

9. Risk field 93: This field displays the level of risk associated with the task from the list of values.

10. Rel. Feature field 95: This field is used to link this task to the features being implemented by the task.

11. **Status field 81:** This field displays the overall status of this task from the list of values.

12. **Design Review field 97:** This field displays the date of the design review for this task.

13. Code Rev field 99: This field displays the date of the code review for this task.

14. **Comp % field 101:** This field displays the portion of the task that has been completed to date.

15. Target Date field 103: This field displays the target date that the task will be completed and code will be unit tested.

In addition, comments may be added and files may be attached to this task

Quality Assurance

QA managers use the Release Manager to manage and track both QA test plans and the tests that are executed against them.

Test Plans

QA managers use the QA Plans List view shown in FIGURE 8 to enter and track information relating to QA test plans developed to test product features. Information about the test plan is entered into the “Plan Description” area, 105, and information about the progress of the effort to develop the test plan is updated in the “Plan Development” area, 107. QA efforts can include, for example, production assembly, inspection, reliability tests, maintainability tests, availability tests, physical tests, performance tests, functional tests, and product specific tests. QA managers may work with product marketing to ensure their test plans are properly linked to the new features they will be verifying. Test plans are entered and displayed into the QA Plans List view, FIGURE 8, in the following fields:

1. Release field 51: Product identification is shown in this field.
2. Title field 53: This field displays a short title for the QA test plan.
3. Description field 55: This field displays a description of the QA test plan.
4. Assoc. Parties field 57: This field associates one or more members of the project team with this test plan
5. QA Group field 109: This field associates one or more QA groups with this test plan.
6. Rel. Feature field 111: This field links the test plan to the features it verifies.
7. Status field 113: This field displays the overall status of this test plan from the list of values.
8. Comp % field 115: This field displays the portion of the effort to develop the QA test plan that has been completed to date.
9. Target field 117: This field displays the target date that the QA test plan will be completed.

10. Passes field 119: This read-only field calculates the number of tests that have been executed against the test plan.

11. Last Build field 119: This read-only field displays the last build number tested with this test plan and the stability status as assessed by the tester during that test.

Tests

QA managers can use the QA Test Details view of FIGURE 9 to enter and track information relating to tests conducted during the course of the release. Information about a test is entered into the "Test Description" area, 131, information about the test environment is entered into the "Test Environment" area, 133, and information about the test results is entered in the "Test Execution Status" area, 135. The following fields may be accessed.

1. Date field 137: The date the test was executed.
2. Tester field 139: Displays the person who conducted the test.
3. Test Plan field 141: Displays the test plan that was executed.
4. Release field 143: This read-only field is determined by the release associated with the test plan selected in step 3.
5. Build field 145: This field displays the build number that was tested.
6. Client OS field 147: This field displays the operating system of the client test machine.
7. Server OS field 149: This field displays the operating system of the server in the test environment.

8. Web Srv. field 151: This read-only field displays the operating system of the web server in the test environment.

9. Database field 153: This field displays the database used for the test from the list of values.

10. Client Type field 155: This field displays the client deployment option (dedicated, Java Thin Client, Thin Client for Windows, or HTML Thin Client) from the list of values.

11. Type field 156: This field displays the test type from the list of values.

12. Status field 157: This field displays the stability status of the tested area from the list of values.

13. Cover % field 135: This field displays the portion of the test plan covered.

14. Pass % field 159: This field displays the portion of the test plan that passed.

Technical Publications

Managers in technical publications use the Tech Documents List view shown in FIGURE 10 to enter and track information relating to documents being developed to describe product features.

Tech pubs managers generally work with product marketing to ensure their documents are properly linked to the features they describe. Documentation records are entered into the Tech Documents List view, FIGURE 10, by filling in the following fields:

1. Release field 51: The product which the feature is a part of.

2. Title field 53: This field displays a short title for the document.

3. Description field 161: This field displays a description of the document.

4. Version field 163: This field displays the version number of the document.
5. Est Pages field 165: This field displays the estimated number of pages of the completed document.
6. New % field 167: This field displays the portion of the document content that will be new once the document is completed.
7. Assoc. Parties field 57: This field associates one or more members of the project team with this particular document
8. Rel. Feature field 53: This links this document to the features it describes
9. Status field 165: This field displays the overall status of this document from the list of values.
10. Comp % field 167: This field displays the portion of the overall effort to develop the document that has been completed.
11. Target field 169: This field displays the target date that the document will be completed.

Additionally, enterprise specific approvals and clearances may be built into the system and method of product release manager. For example, senior management and executive approvals, and “sign-off’s” by advertising, marketing, sales, distribution, channel partners, business practices, legal, insurance, product safety, and environmental approvals may be incorporated into the system and method described herein.

Relational Database Management System for Implementing The Release Manager Method and System

FIGURES 11 and 12 illustrate database schema for a simple relational database management system for use with method and system of the release manager. The matrices 1101, 1102, 1103, 1104, 1105, 1106, and 1107 represents a particular entry type, as PROJECT, FEATURE, ENGINEERING, QUALITY ASSURANCE, QUALITY ASSURANCE (INDIVIDUAL TEST), TECHNICAL PUBLICATIONS and ENTERPRISE SPECIFIC. The first bolded cell represents the master key, as well as the entry for the project id, which is assumed to be unique. The second bolded cell contains the feature id, which when concatenated with the project id is assumed to be a unique candidate key for the feature. Note that "keys" are equivalently referred to as integrity constraints.

New tasks and tests are created by initiating an appropriate INSERT operation and by initiating the appropriate MODIFY operations, as is well known in the relational database art. Screens and views are provided by the appropriate SELECT and FILTER operations, typically in association with view or form applets.

The functional stages and steps, engineering, quality assurance, and technical publications, as well as enterprise specific activities are illustrated, where the concatenation of the project id, the feature id, and a function specific id provides a candidate key, integrity constraint, or foreign key. As shown in FIGURE 12, this candidate key can be used to retrieve all of the activities relating to a feature, for example, by searching on the concatenation of PROJECT ID and FEATURE ID, optionally with functional area entries (as for engineering, quality assurance, technical publications, and other ancillary functions) properly set so as to retrieve, for example, quality assurance activities with respect to a particular feature.

The above description is strictly illustrative, and other database management systems, schema, and strategies may be used without departing from the spirit of the invention.

Managing the Release

The Release Manager is more than a helpful tool for product marketing, engineering, quality assurance, technical publications and corporate staff functions to track release-related activities.

It is also useful in managing the release and ensuring that important details don't fall through the cracks. Following are examples of how the Release Manager might be used to ensure an on-time release, a high quality, and customer satisfaction:

Product marketing can ensure that important features are not forgotten by querying the Release Manager regularly for features associated with contractual requirements, high revenue customers, or high-profile customers.

Since candidate features are maintained in the Release Manager from the moment they are first considered, engineering can get an early start on scoping development effort and need not wait for a formal release plan.

Product marketing can ensure that all features are properly specified in Marketing Requirements Documents. Features without matching Marketing Requirements Documents are immediately apparent.

The most current version of engineering, marketing, and technical documents is always available in Siebel Release Manager.

Engineering can allocate development resources efficiently based on the relative priority of the features to which engineering tasks are linked.

QA managers reduce quality risk by focussing testing efforts on those test plans that are associated with the greatest number of new features.

Tech pubs and QA can guarantee release integrity by ensuring that all features are properly documented and tested. Features not linked to any documents or test plans will be immediately apparent in the Feature List View of the Release Manager.

Release meetings will be run more efficiently. Since status is available to the entire organization through the Release Manager, valuable time need not be devoted to covering this information in detail at release meetings.

Activities may be linked to features, engineering tasks, QA test plans, and tech docs. For example, a multi-value link allows easy setup of project teams, tech pubs documents, and test plans for new releases based on a previous release. Similarly, linking of defects to test plans may be supported, as in associating a test plan with a defect.

The Priority control in the Engineer Task List view may be configured to display the highest priority of any features linked to it.

An explorer view provides convenient display of all engineering task, test plans, and tech pubs documents associated with a given feature.

While the invention has been illustrated and described with respect to certain preferred embodiments and exemplifications, it is not intended to limit the scope of the invention thereby, but solely by the claims appended hereto.